Acessibilidade e Vulnerabilidade em Aplicações Android

Bruno Barbosa Torres (bbt)

José Murilo Mota (jmsmf)

AccessiLeaks: Investigating Privacy Leaks Exposed by the Android Accessibility Service

Mohammad Naseri, Nataniel P. Borges Jr.,
Andreas Zeller, Romain Rouvoy

- Accessibility service
- Attack model and vulnerabilities
- The article's approach
  - AcDetect
  - AcFix
  - AcGuard
- What can be done?
- Wrapping up

Accessibility service

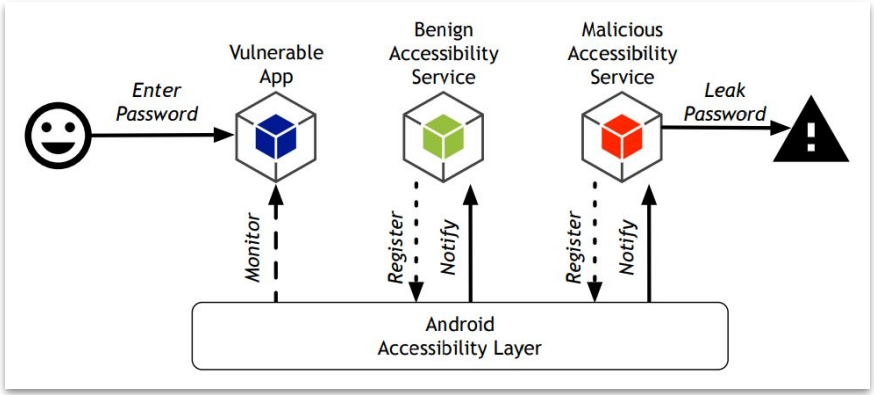
Accessibility Service

- Standard since Android version 1.6 - Donut (API 4);
- Capabilities include:
  - Alternative Navigation feedbacks
  - Text-to-speech conversion
  - Haptic feedback
- Using it allows the app to monitor screen content;
- Good: Allows Devs to make their own accessibility tools to help the user;
- Bad: Easily exploitable by bad actors and hackers.

Attack model and vulnerabilities

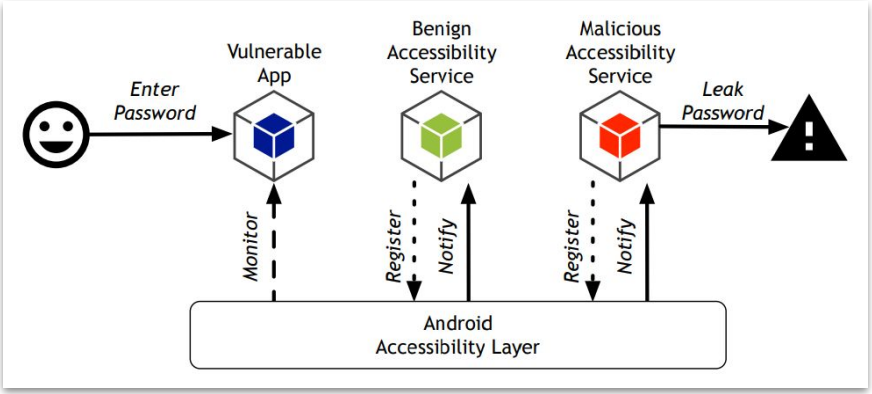
Attack model and vulnerabilities

- Main vulnerability: Eavesdropping;
- 3 Components needed:
  - **Enabled Accessibility**
  - Malicious service listener
  - A vulnerable app
- By default, accessibility is turned off, requiring explicit user consent to be enabled;
- Once an event is triggered, listeners of that type of event are notified;
- 25 Types of accessibility events (*View clicked*, *View text changed*, *Window content changed*...).



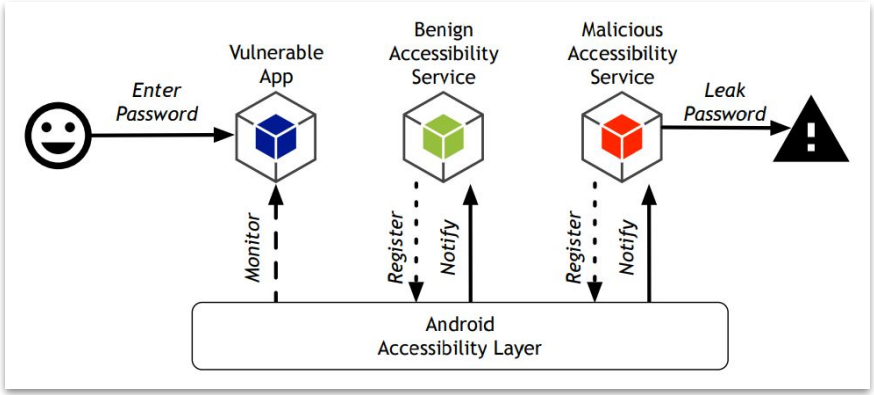
Attack model and vulnerabilities

- Main vulnerability: Eavesdropping;
- 3 Components needed:
  - Enabled Accessibility
  - **Malicious service listener**
  - A vulnerable app
- To exploit the Accessibility Service the attacker doesn't need to request any specific capabilities since *TextChanged* events are always fired.



Attack model and vulnerabilities

- Main vulnerability: Eavesdropping;
- 3 Components needed:
  - Enabled Accessibility
  - Malicious service listener
  - **A vulnerable app**
- Every time an User enters text in an *EditText* field, an event is generated and forwarded to all registered services;
- Passwords only have the last inputted letter available;
- To avoid this, *importantForAccessibility* has to be set anything other than *yes* or *auto* (default is *auto)*.



The article's approach

The article's approach

The article's approach to this widespread problems are three:

- How to detect if an app publicly available is vulnerable (AcDetect);
- How to help developers easily fix their code (AcFix);
- How to help users be aware of what app may eavesdrop on their actions (AcGuard).

AcDetect

AcDetect

A tool to analyze an Android app binary for accessibility service vulnerabilities

README.md

AcDetect

AcDetect is a tool developed to help app stores identify if an app has an accessibility service privacy vulnerability.

Prerequisites

What things you need to install to use AcDetect.

```
- Python3
- ApkTool
```

Execution

```
python main.py sample.apk
```

AcDetect

- Extract all resources from the APK using APKTool. (user defined strings, layout files...)
- Parses the resources to locate vulnerable input fields.
  - An input field is considered vulnerable if it is input field, contains a password and is important for accessibility.

```
<EditText
        android:id="@+id/plain_text_input"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:inputType="textPassword"
<!--          android:importantForAccessibility="yes"-->
<!--          android:importantForAccessibility="auto"-->
```

- Does not inspect the apps compiled source code for elements created at runtime, as well as for *input type* and *important for accessibility* assignments.

AcDetect

- Used AcDetect to automatically detect vulnerabilities on all 100 apps and manually discarded vulnerabilities found outside of the apps main login screen.

Finance Category

#	App	AcDetect	Manual
1	at.paysafecard.android	Y	Y
2	com.paypal.android.p2pmobile	Y	Y
3	com.starfinanz.smob.android.sfinanzstatus	Y	N
4	com.starfinanz.mobile.android.pushtan	N	N
5	de.check24.check24	Y	N
6	com.db.pbc.phototan.db	N	N
7	de.ingdiba.bankingapp	N	Y
8	com.westernunion.moneytransferr3app.eu	Y	Y
9	com.google.android.apps.walletnfcrel	Y	Y
10	de.commerzbanking.mobil	Y	Y
11	de.fiducia.smartphone.android.securego.vr	N	N
12	com.db.pwcc.dbmobile	Y	Y
13	de.postbank.finanzassistent	Y	Y
14	de.dkb.portalapp	N	Y
15	com.libertex.mobile	N	N
16	de.wirecard.boonpayment	Y	Y
17	de.sdvrz.ihb.mobile.secureapp.sparda.produktion	Y	Y

Social Category

#	App	AcDetect	Manual
1	com.instagram.android	Y	Y
2	com.snapchat.android	Y	Y
3	com.facebook.katana	Y	Y
4	com.pinterest	Y	Y
5	net.lovoo.android	Y	Y
6	com.badoo.mobile	Y	Y
7	com.facebook.lite	N	Y
8	com.linkedin.android	Y	Y
9	com.tumblr	Y	Y
10	com.jaumo	Y	Y
11	com.facebook.Socal	N	N
12	com.herzick.houseparty	Y	Y
13	de.nebenan.app	Y	Y
14	com.narvii.amino.master	Y	Y
15	de.startupfreunde.bibflirt	Y	Y
16	com.GermanyChatMessenger	Y	Y
17	com.enterkomug.linduu	Y	Y

AcDetect (evaluation)

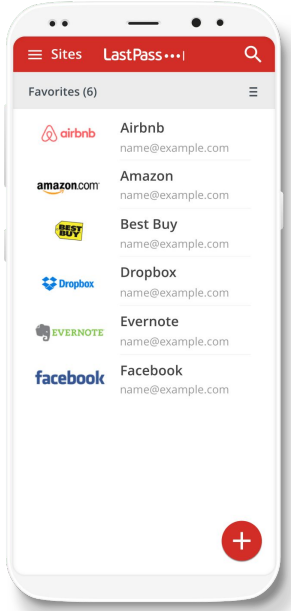
	Classified as			
Input	**True**	**False**	**Total**	
True	TP = 71	FN = 5	76	*Precision* = 89 %
False	FP = 9	TN = 15	24	*Recall* = 93 %
Total	80	20	100	*Accuracy* = 86 %
				Specificity = 63 %

AcDetect *detected accessibility vulnerabilities in the top 50* Finance *and* Social *apps with a precision of 89 % and a recall of 93 %.*

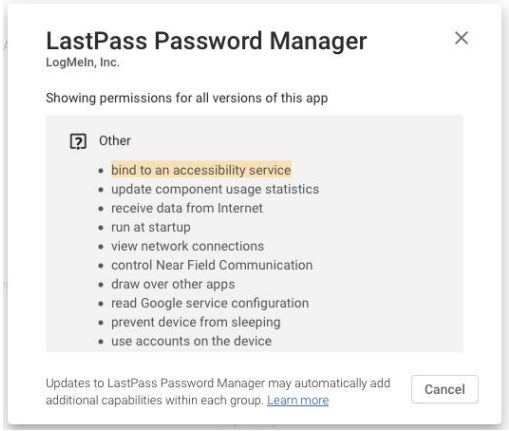
AcDetect (discussion)

- The Android developer's guide determines that *"accessibility services should only be used to assist users with disabilities in using android devices and apps"*;
- Developers have used this service to provide distinct functionality;
- Large number of apps using *accessibility services* for, among other functionalities, password management;
- The article's *key logger* app stayed in the Play Store for 4 months and received no notification.

> ★ **Note:** Although it's beneficial to add accessibility features in your app, you should use them only for the purpose of helping users with disabilities interact with your app.

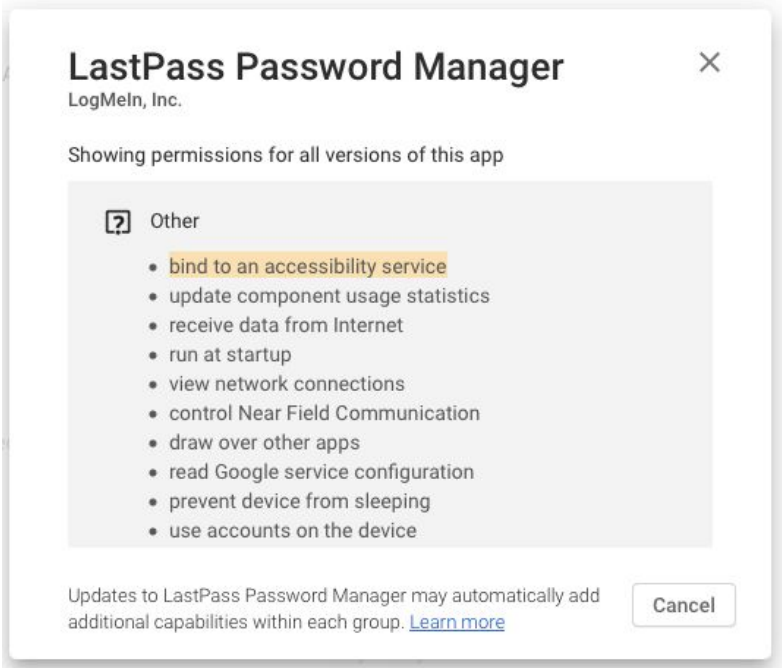


LastPass •••|



AcDetect (discussion)

- To listen to *accessibility events* the developer must request the `android.permission.BIND_ACCESSIBILITY_SERVICE` permission in the app' manifest;
  - If request at the application level, it is displayed in the app download page in the Google Play Store;
  - If requested only at the service level (which is mandatory by the Android Framework), then no record of the permission is mentioned from Google Play
- 55.8% of the accessibility apps do not notify the users about their accessibility service use.



AcFix

AcFix

- A tool for developers that finds vulnerabilities in their code, and quickly fix them in 3 steps;
- First, it analyzes the source code of the Android app;
- Second, it spots the occurrences of vulnerable code;
- Third, attempts to fix them.

AcFix (technique)

1. It generates the **project's source code AST** using an extension (Spoon);
2. It uses the AST to process all layout files and Java classes in order to **find all the password text fields**, covering all forms of bindings;
3. Reuses the password text field definition from AcDetect;
4. For each candidate field, it checks all possible ways of **assigning the *importantForAccessibility* attribute**, if there's an attribution, it changes the attribute directly for static fields, or inserts a *setImportantForAccessibility* function for dynamic fields.

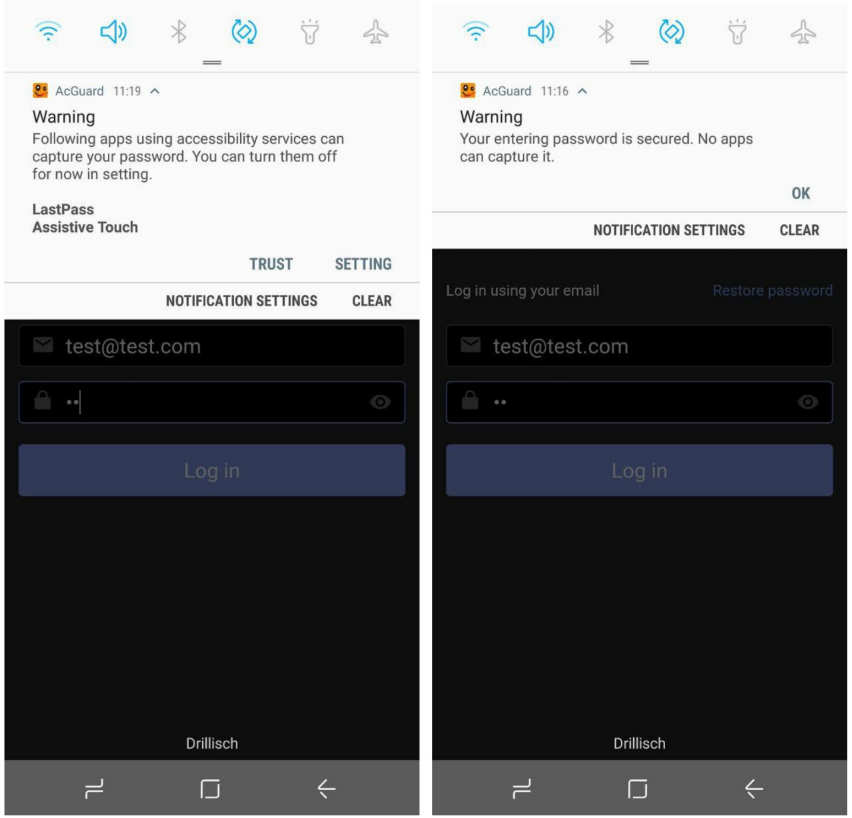
AcFix (evaluation)

- For the 40 open source vulnerable apps that AcFix was tested on, it successfully fixed all 40;
- At the time of publishing, 70% (28 apps) had *merged* the *pull request* that contained the fix;
- The remaining 25% (10 apps) had no answer, indicating that the projects aren't any longer under maintenance.
- 5% (2 apps) raised questions about reduced accessibility after fix implementation and didn't *merge* the fix;

AcGuard

AcGuard

- Android app;
  - Monitors the user interactions and notifies users about potential threats before they enter a password;
- Implements an accessibility service without any specific capability;
- When a password field is detected, AcGuard queries the OS for the the list of enabled accessibility services and alerts the user accordingly.



AcGuard (evaluation)

- "How the warning notifications provided affected the users, as well as identifying who the users would expect to fix the problem"
- Online survey
  - 50 Android users
  - Ages 18 to 30
  - University students
- Scenario
  - Banking app on their own personal and updated Android phone;
  - Provided 3 screenshots of the finance app's login screen:
    - Without any notification;
    - "No apps can read your password" notifications from AcGuard;
    - Warning informing the users that some apps on their phone could be reading their passwords.

Table 4. Average and standard deviation of confidence level values of the three user study questions results

Snapshot State	Median	Average	Standard Deviation
No Notification	5	5.3	1.8
Secure Notification	7	6.3	2.5
Warning Notification	3	3.8	2.3

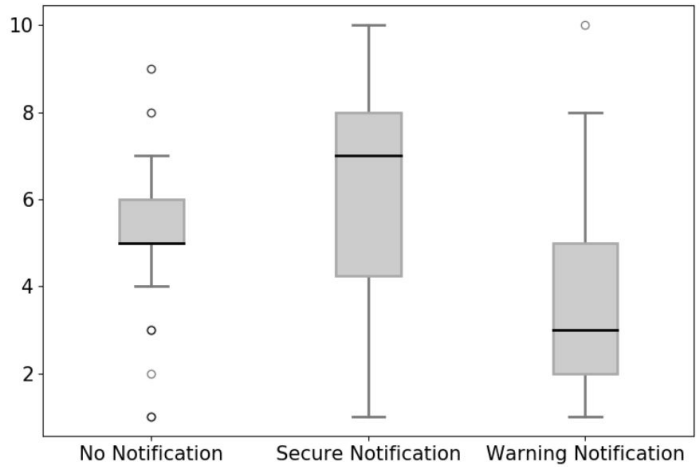


Fig. 6. Users' confidence level for three notification states

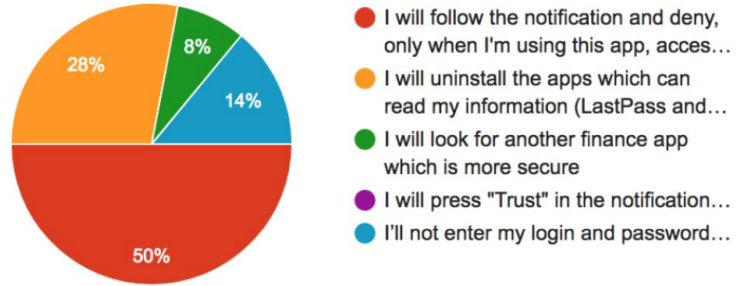


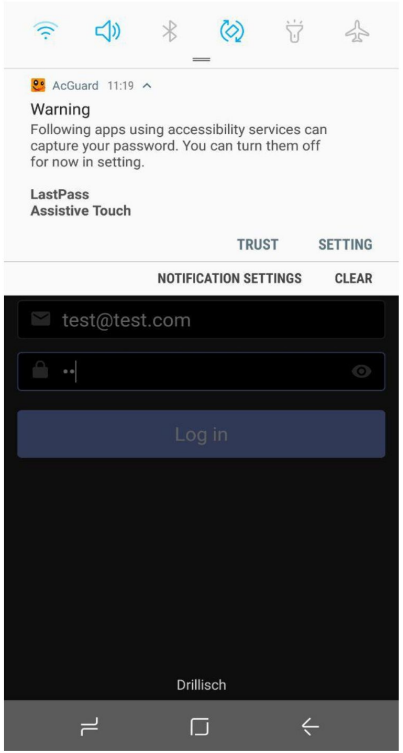
Fig. 7. User responses

With its notifications, AcGuard *could affect the users' confidence in an app. 50 % of users followed the* AcGuard *recommendation.*

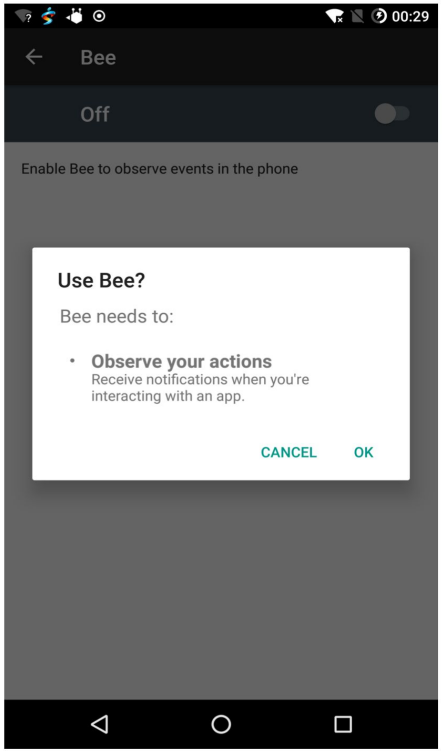
What can be done

(by Google)

Native notification. While AcGuard helps users to identify and react against malicious apps using accessibility services, increasing or decreasing their confidence to enter their password in an app, it is still necessary that user installs and trusts it, as it exploits the accessibility service vulnerability to warn user of potential threats. A notification similar to AcGuard could be provided by the OS, for a widespread mitigation of the vulnerability.



User Interface. To enable our app's accessibility service, Android warns users with the message displayed in Figure 8. However, this message—provided by the Android operating system—is uninformative regarding possible vulnerabilities, including the possibility of capturing user inputs while typing passwords. For most users, this notification seems legit, thus they usually confirm it. Studies, such as [15], illustrate the importance of meaningful messages.



Impact on Users with Disability. AcFix and AcGuard are the tools that we developed to mitigate the existing accessibility vulnerability. Nevertheless our solutions can impact people with disability. AcFix, which is specifically designed to fix password vulnerability, disables accessibility feature for password inputs. This may trouble users with disabilities, such as people who are blind or have low vision, who are unable to see screens and hence cannot use touchscreen keyboards or users with dexterity problems in using their fingers. That is the reason why we emphasized that AcFix is a quick solution for apps in the wild to be protected against such attacks. Google has provided accessibility services, such as *TalkBack*, *Select to Speak* and *Text-to-speech*[20] to help users with aforementioned disabilities. Such services are integrated in Android operating system. However, those services are not widely used by users. For instance, Rodrigues et al. [23] showed the usability of *TalkBack* is pretty daunting for users with visual impairments. Naftali and Findlater [19] found that users with motor impairments have challenges using services like *Select to Speak*. Although we can not generalize such experiments, it shows that the provided services are not widely used among users with disabilities and the current accessibility vulnerability in the wild is more severe. AcGuard does not impact users with disability as it does not enforce the user to disable any accessibility service. It just warns the user and usually people with disabilities are familiar with the services that they use occasionally. In the following, we would like to discuss the operating system changes that would achieve both accessibility and privacy of user data.

OS Changes. For the Android operating system, based on our studies, we can formulate some suggestions that would contribute to improve the security of the current model in terms of the accessibility services:

- **Permission Model.** In the current model, there is only one permission `BIND_ACCESSIBILITY_SERVICE` to request access to the accessibility service, which allows the app to receive all the accessibility events, no matter the app is in background or foreground. This issue is exploited by malicious apps, which listen and receive the events including information about other apps. A finer-grained permission model, with distinct permissions for accessibility services: *i)* a permission to receive accessibility notifications only from the foreground app, which can be considered as benign since it only receiving events regarding their own app, and *ii)* one permission for apps running in the background apps, which could be easily identified by users and would be required to provide a comprehensive description explaining their usage of the service before publication.

- **Predefined Accessibility Services.** As we explained before, Google has provided predefined set of accessibility services to help users with disability which developers can utilize in their application. The problem is that there is not difference between those services and new services, which are declared by programmers in current operating system structure. Once an event is fired, all enabled services receive the event. Since predefined services are hugely used by users with disabilities, a better solution would consist in separating the event's type of these two. In that case, developers can utilize predefined accessibility services in their app, such as password fields, without worrying about other services receiving triggered events.

Wrapping up

Wrapping up

- Extension of the developed tools;
- Threats to validity;
- Related work:
  - Accessibility services;
  - Dangerous permissions;
  - Developer support;
- Future work:
  - Acquire more sensitive information (such as payment data);
  - Incorporate more advanced static analysis techniques to handle dynamically created and updated widgets.
- Link para o artigo

"How can a platform be **open for accessibility** services, and at the same time, be **closed for possible abuses** of the associated APIs?"

É isso aí

Dúvidas?